



Docket No.: 449122063900
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Karlheinz KRAUSE et al.

Application No.: 10/668,603

Filed: September 24, 2003

Group Art Unit: Unassigned

For: PROGRAMMING FLASH MEMORY VIA
BOUNDARY SCAN REGISTER

Examiner: Unassigned

CLAIM FOR PRIORITY AND SUBMISSION OF DOCUMENT

MS Patent Application
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

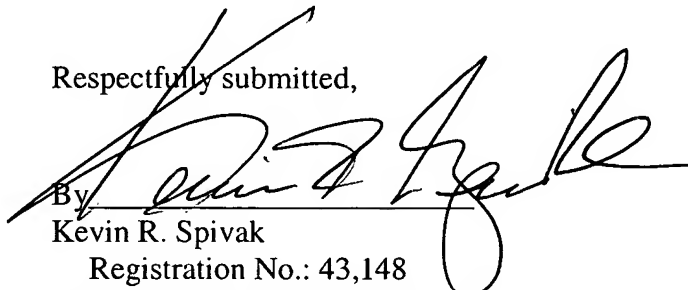
Applicant hereby claims priority under 35 U.S.C. 119 based on the following prior foreign application filed in the following foreign country on the date indicated:

<u>Country</u>	<u>Application No.</u>	<u>Date</u>
Germany	102 44 757.8	September 25, 2002

In support of this claim, a certified copy of said original foreign application is filed herewith.

Dated: February 17, 2004

Respectfully submitted,

By 
Kevin R. Spivak
Registration No.: 43,148

MORRISON & FOERSTER LLP
1650 Tysons Blvd, Suite 300
McLean, Virginia 22102
(703) 760-7762 – Telephone No.
(703) 760-7777 – Facsimile No.



1

—

BUNDESREPUBLIK DEUTSCHLAND

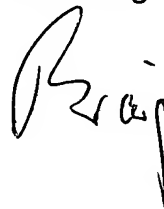


Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen: 102 44 757.8
Anmeldetag: 25. September 2002
Anmelder/Inhaber: Siemens Aktiengesellschaft,
München/DE
Bezeichnung: Programmierung eines Speicherbausteins über ein
Boundary Scan-Register
IPC: G 11 C 16/10

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 25. September 2003
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag



Brosig

Beschreibung

Programmierung eines Speicherbausteins über ein Boundary Scan-Register

5

Die zugrunde liegende Erfindung basiert auf einem Verfahren sowie einer Schnittstelle zur On-Board-Programmierung und/oder In-System-Konfiguration (ISC) eines auf einer Leiterplatte montierten Speicherbausteins durch Ansteuerung seiner Eingänge mit Hilfe eines auf derselben Leiterplatte montierten anwendungsspezifischen integrierten Halbleiterbausteins über ein Boundary Scan (BSCAN)-Register, dessen Ausgangssignale zur Aktivierung bzw. Deaktivierung einer Schreiboperation vorgesehen sind.

15

Die Boundary-Scan-Technik (BSCAN) ist ein standardisiertes Verfahren für Boardtests, das 1990 formell als Industriestandard IEEE 1149.1 für Test Access Port (TAP)- und Boundary Scan (BSCAN)-Architekturen verabschiedet wurde. Auf dieser Spezifikation basieren Verbindungstests auf Boardebene in der Produktion von komplexen Printed Circuit Boards (PCBs). Besitzt der Prüfling einen eigenen Mikroprozessor nebst Flash-basiertem Programmspeicher, kann ein Built-In Self Test beispielsweise durch Laden des Flash-Speichers via Boundary Scan mit Hilfe eines Selbsttestprogramms implementiert werden. Im Speicher abgelegte Testresultate sind dabei nach Beendigung des Testvorganges wiederum per Boundary Scan auslesbar.

20

25

30

Zur Durchführung von Boundary Scan-Tests müssen zwei Bedingungen erfüllt sein: Zumindest einige der integrierten Schaltungen (ICs) auf der Platine müssen der BSCAN-Spezifikation entsprechen. Bei der Prüfung wird dann mit Hilfe von Testvektoren ein BSCAN-Register dazu veranlasst, den gewünschten

Test auszuführen. Darüber hinaus müssen die Produktentwickler einen Scanpfad zwischen den einzelnen ICs zur Verfügung stellen, der von einem Test Access Port (TAP) durch die ICs hindurch wieder zurück zum TAP führt, wo die Daten schließlich abgetastet werden. Beim Prüfen von elektrischen Anschlüssen stellen Boundary Scan-Tests eine ausgezeichnete Alternative zu In-Circuit-Tests (ICTs) dar. Die Kosten für die Durchführung der Funktionsprüfung sind niedrig, und aufgrund der zunehmenden Miniaturisierung integrierter Halbleiterbauteile ist anzunehmen, dass sich der Trend hin zu Boundary Scan weiter fortsetzen wird.

Der IEEE-Standard 1149.1 spezifiziert neben den notwendigen Hardwarestrukturen auch eine adäquate Beschreibungsform all dieser Eigenschaften in Form der Boundary Scan Description Language (BSDL). Dabei ist dieser Standard so offen gehalten, dass durch Definition kundenspezifischer Register und entsprechender Instruktionen applikationsspezifische Funktionen implementiert werden können, ohne an Kompatibilität zu verlieren. Genau auf diesem Umstand beruhen prinzipiell auch alle konventionellen Verfahren zur In-System-Programmierung (ISP) integrierter Halbleiterbauteile, die auf der Basis von Boundary Scan arbeiten, allerdings existierte dafür bisher kein Standard. Auch eine einheitliche Definition von Testvektorformaten zum Datenaustausch ist im IEEE-Standard 1149.1 nicht enthalten.

Wurde das Boundary Scan-Verfahren gemäß IEEE-Standard 1149.1 bisher vor allem als innovative Technologie zur Funktionsprüfung integrierter Schaltungen bzw. zur Verifikation und Simulation von Hardwarefehlfunktionen verwendet, zeigen jüngste Entwicklungen weitere Einsatzmöglichkeiten dieses Prinzips auf. Neben der Nutzung zu Testzwecken wird Boundary Scan auch

sehr effektiv im Bereich der sogenannten In-System-Programmierung (ISP) von Flash-Speichern sowie Programmable Logic Device (PLD)-Bausteinen, wie z.B. Field-Programmable Gate Arrays (FPGAs) mit bis zu 10.000 Logik-Gattern pro Array oder

5 Programmable Logic Arrays (PLAs), eingesetzt. Dabei werden die einzelnen Steuer- und Adress-Eingänge eines Flash-Speichers über die diesen Eingängen zugeordneten, miteinander verketteten BSCAN-Zellen eines BSCAN-Registers derart stimuliert, dass wahlweise eine Lese- oder Schreiboperation ausgelöst wird. Wie aus der in Fig. 1 abgebildeten Prinzipskizze zu entnehmen ist, können die Steuer-, Adress- und Datensignale von den entsprechenden BSCAN-Zellen erfasst und ausgegeben werden.

15 Mit dem im Januar 2001 herausgegebenen IEEE-Standard 1532 wurde erstmals eine einheitliche Normierung für die Systemarchitektur und ein geeignetes Datenformat zur In-System-Konfiguration (ISC) von programmierbaren integrierten Halbleiterbauteilen geschaffen. Dieser Standard beschreibt eine
20 Reihe von obligatorischen und optionalen Programmierinstruktionen und entsprechenden Datenregistern. Prinzipiell stellt der IEEE-Standard 1532 eine auf die speziellen Anforderungen der Programmierung abgestimmte Erweiterung des IEEE-Standards 1149.1 zur Standardisierung des Programmierprozesses für programmierbare Halbleiterbausteine dar, ist aber zu diesem
25 vollständig kompatibel. Da sich der IEEE-Standard 1532 jedoch ausschließlich auf programmierbare Halbleiterbausteine bezieht, die eine JTAG-Schnittstelle aufweisen und in der Lage sind, Programmierdaten intern abzuspeichern, bezieht sich
30 dieser Standard nicht auf Flash-Speicher ohne JTAG-Schnittstelle. Grundsätzlich umfasst der IEEE-Standard 1532 die Standardisierung spezieller ISC-Datenregister, ISC-Instruk-

tionen, BSDL-Erweiterungen für die Beschreibung der ISC-Merkmale sowie ein spezielles Dateiformat.

5 Damit geht der Standard IEEE 1532 weit über alle bisherigen
Lösungen hinaus und ermöglicht auf Basis seines innovativen
Charakters auch die simultane Programmierung mehrerer kompa-
tibler Bauelemente. Durch eine Reihe von zusätzlichen Funkti-
onen, wie z.B. der Nutzung spezieller Programmier-Spannungs-
Pins, Compliance-Enable-Pins und der Möglichkeit einer Defi-
10 nition optionaler ISC-Instruktionen, bietet IEEE 1532 auch
die notwendigen Freiräume zur Schaffung applikationsspektifi-
scher Kompatibilitäten. Beim IEEE-Standard 1532 erfolgt dabei
grundsätzlich eine methodische Trennung von Prozessinformati-
onen und Programmierdaten.

15

Die für die Programmierung notwendigen Bauelementeinformatio-
nen sind in entsprechenden „BSDL-Erweiterungen“ eines BSDL-
Files enthalten. Ein aus Nutzersicht ganz wesentlicher Be-
standteil ist hierbei der sogenannte „Attribute ISC_Flow“.
20 Dieser nimmt die Umsetzung von Basisprogrammierungsfunktionen,
wie z.B. Löschen, Programmieren, Verifizieren etc., in ent-
sprechende Teilsequenzen vor. Sollen mehrere IEEE-1532-
kompatible Bauelemente gleichzeitig programmiert werden, muss-
die Programmiersoftware die Fähigkeit besitzen, eine beliebi-
25 ge Anzahl von „Attribute ISC_Flows“ virtuell zusammenzufüh-
ren. Aufgrund dieser Methodik bietet der IEEE-Standard 1532
die Möglichkeit, PLDs verschiedener Hersteller unabhängig von
der Prozesstechnologie (z.B. EEPROM-, SRAM- oder Flash-
basiert), ihrer Architektur oder ihrer Spannungspegel zu pro-
30 grammieren.

Die BSDL-Files müssen typspezifisch vom jeweiligen Chipher-
steller bereitgestellt werden. Sie sind insofern ein ganz we-

sentlicher Teil des Lieferumfanges. Demgegenüber werden die Programmierdaten individuell durch den PLD-Designer über einen entsprechenden Target-Compiler in Form eines Daten-Files erzeugt. Ohne auf nähere Einzelheiten einzugehen, sei nur erwähnt, dass es sich hierbei um ASCII-Dateien mit einer speziellen Syntax handelt, so dass sie gelesen und editiert werden können. Für jedes zu programmierende Bauelement müssen nach diesem Funktionsprinzip ein BSDL-File und ein dazu konsistentes Daten-File vorhanden sein.

10

Auch wenn die bisher beschriebene Theorie des IEEE-Standards 1532 vielversprechend erscheint, reicht das für den praktischen Erfolg dieses Standards keinesfalls aus. Insbesondere die Programmiersoftware nimmt aufgrund der Vielzahl notwendiger Funktionen eine Schlüsselstellung ein und bestimmt maßgeblich die Effizienz der ISC-Operationen.

20

Bei der simultanen Programmierung mehrerer Bauelemente hängt die erzielbare Zeiteinsparung von vielen verschiedenen Faktoren ab. Dabei sind insbesondere die Komplexität, Architektur, Technologie und Taktrate des betreffenden Bauelements zu nennen. Insofern sind verifizierbare quantitative Aussagen darüber abhängig von der jeweiligen Implementierung und Applikation.

25

Gerade bei Boundary Scan spielt die kontinuierliche Weiterentwicklung der Softwarewerkzeuge eine entscheidende Rolle für die effektive Umsetzung dieser zukunftsweisenden Technologie in der Praxis. In Zusammenhang mit der BSCAN-basierten On-Board-Programmierung von Speicherbausteinen ist dabei als Stand der Technik insbesondere die integrierte Entwicklungs- und Programmierungsumgebung SYSTEM CASCON™ der Herstellerfirma GÖPEL electronic GmbH aus Jena zu erwähnen.

30

Um die neue Programmiermethode mit anderen BSCAN-Prozeduren wie Debugging, Produktionstests oder Flash-Programmierung zu kombinieren, ist bei GÖPEL ein J-Drive Programming Engine vorgesehen, das zur In-System-Konfiguration (ISC) eines PLDs direkt in die Boundary Scan-Softwarepakete SYSTEM CASCON™ und POLARIS™ eingebunden ist. Dieses J-Drive Programming Engine greift dabei auf Konfigurationsdaten aus einem BSDL-File zur Konfiguration eines Test Access Ports (TAP) gemäß IEEE-Standard 1149.1 zu, welches als Steuereinheit zur Ansteuerung der BSCAN-Zellen des BSCAN-Registers über eine Programmierschnittstelle verwendet wird. In Verbindung mit den zur Verfügung stehenden Controllern auf Basis von USB, PCI, PXI oder VXI lassen sich dadurch Multi-Mode-Boundary-Scan-Systeme mit zugeschnittener Leistungsfähigkeit für Labor, Produktion und Service kostenoptimiert konfigurieren. Auf diese Weise wird es möglich, beispielsweise bei der Produktion mit nur einem Gerät ein auf einer Leiterplatte montiertes PLD auf Fertigungsfehler zu testen, anschließend das PLD zu konfigurieren und eine bestimmte Produktionsversion der Firmware in einen vorhandenen Flash-Speicher zu laden.

Nach dem Stand der Technik erfolgt die Programmierung von Flash-Speichern in der Regel mit Hilfe von anwendungsspezifischen integrierten Halbleiterbausteinen (ASICs), die in der Lage sind, den betreffenden Flash-Speicher auch während des Betriebs zu konfigurieren. In der Regel sind dies ASICs, die den JTAG-Standard IEEE 1149.1 unterstützen (vgl. Fig. 1: Schnittstelle zwischen ASIC-1 und Flash-1). Zu diesem Zweck wird ein ggf. verkürztes BSCAN-Register benutzt, mit dessen Hilfe die für die Programmierung des Speicherbausteins benötigte Zeitdauer entscheidend verkürzt werden kann.

Bei Anwendung dieses Verfahrens ergibt sich jedoch eine Reihe von Problemen: So ist beispielsweise zur Programmierung eines Flash-Speichers die Erstellung eines Programms verhältnismäßig aufwendig, da die Konfiguration (d.h. die Verbindungen zwischen ASIC und Flash-Speicher) aus dem Stromlaufplan bzw. aus der daraus ableitbaren Netzliste entnommen und das BSDL-File des ASICs herangezogen werden muss. Ferner ist eine simultane Programmierung mehrerer, auf derselben Leiterplatte montierter integrierter Halbleiterbausteine derzeit nicht möglich. Darüber hinaus ist bei herkömmlichen Verfahren auch ein Burstbetrieb nicht möglich, da bei jedem Schreibzyklus die Programmierdaten in den ASIC geschoben werden müssen sowie in der Regel auch die Adressen und die Steuerbits.

15 AUFGABE DER VORLIEGENDEN ERFINDUNG

Ausgehend von dem oben genannten Stand der Technik, ist die vorliegende Erfindung der Aufgabe gewidmet, ein Verfahren zur On-Board-Programmierung sowie In-System-Konfiguration (ISC) eines Speicherbausteins durch Stimulierung einzelner seiner Eingänge über Speicherzellen eines BSCAN-Registers zur Aktivierung bzw. Deaktivierung einer Schreiboperation bereitzustellen, bei dessen Anwendung der für die On-Board-Programmierung benötigte Stromlaufplan bzw. die daraus ableitbare Netzliste nicht mehr herangezogen werden muss.

Weiterhin soll die simultane Programmierung mehrerer, auf derselben Leiterplatte montierter Speicherbausteine auch bei Burstbetrieb ermöglicht werden.

Diese Aufgabe wird erfindungsgemäß durch die Merkmale der unabhängigen Patentansprüche gelöst. Vorteilhafte Ausführungs-

beispiele, die den Gedanken der Erfindung weiterbilden, sind in den abhängigen Patentansprüchen definiert.

ZUSAMMENFASSENDE DARSTELLUNG DER VORLIEGENDEN ERFINDUNG

5

Im Rahmen der erfindungsgemäßen Lösung ist zur Lösung der im vorangehenden Abschnitt definierten Aufgabe ein Verfahren sowie eine parallele Schnittstelle zur On-Board-Programmierung und/oder In-System-Configuration (ISC) eines auf einer Leiterplatte montierten Speicherbausteins durch Ansteuerung einzelner Eingänge des Speicherbausteins mit Hilfe eines auf derselben Leiterplatte montierten anwendungsspezifischen integrierten Halbleiterbausteins über einzelne Speicherzellen eines BSCAN-Registers zur Aktivierung bzw. Deaktivierung einer Schreiboperation vorgesehen. Dabei ist die Architekturbeschreibung des Halbleiterbausteins und des zu programmierenden Speicherbausteins sowie das Datenformat der Programmier- (DATA_IN) und Konfigurationsdaten (Conf data) in einem BSDL-File abgespeichert. Die Leiterplatte ist über eine zur Durchführung einer Funktionsprüfung des Speicherbausteins geeignete JTAG-Schnittstelle zur Ein- bzw. Ausgabe von Standardbus-signalen (TDI, TDO, TMS, TCK und TRST) sowie zur Eingabe der Ansteuersignale des Halbleiterbausteins ansteuerbar.

25

Zur Verringerung des Programmieraufwands werden erfindungsgemäß die Daten des Stromlaufplans bzw. der daraus ableitbaren Netzliste, welche die Konfiguration der Schnittstelle zwischen dem zu programmierenden Speicherbaustein und dem Halbleiterbaustein festlegen, in einer weiteren Datei (BSDL-File) abgespeichert, so dass der zu programmierende Speicherbaustein nach Anpassung der Eigenschaften des Halbleiterbausteins an den IEEE-Standard 1532 über eine JTAG-Schnittstelle

30

in gleicher Weise wie ein programmierbarer Logikbaustein (PLD) programmiert werden kann.

KURZBESCHREIBUNG DER ZEICHNUNGEN

5

Weitere Eigenschaften, Merkmale, Vorteile und Anwendungen der zugrunde liegenden Erfindung ergeben sich aus den untergeordneten abhängigen Patentansprüchen sowie aus der folgenden Beschreibung zweier Ausführungsbeispiele der Erfindung, welche

10

in Fig. 1 bis 3 dargestellt sind. Hierbei zeigt

Fig. 1 ein Schaltungslayout eines Printed Circuit Boards (PCBs), welches zur Programmierung und Konfiguration zweier Flash-Speicher über ein Boundary Scan (BSCAN)-Register via JTAG-Schnittstelle nach dem Standard IEEE 1532 geeignet ist, wobei die Schnittstelle zwischen dem ASIC und dem zweiten Flash-Speicher durch einen Teil des BSCAN-Registers gebildet wird,

Fig. 2 ein reduziertes Schaltungslayout desselben PCBs zur Programmierung und Konfiguration eines integrierten Flash-Speichers über ein BSCAN-Register via JTAG-Schnittstelle auf Basis eines BSDL-Files nach dem Standard IEEE 1532, wobei ein Configuration-Register gemäß IEEE-Standard 1532 mit dem Ausgang eines anwendungsspezifischen Halbleiterbausteins verbunden ist, und

Fig. 3 Detailansichten zweier Ausführungsbeispiele des Schaltungslayouts zur Verschaltung der Multiplexer und Flipflops einer Scan-Zelle des BSCAN-Registers und einer Capture-Zelle des Configuration-Registers

zur Bereitstellung der für die Programmierung des Flash-Speichers benötigten Steuer-, Daten- und Adresssignale.

DETAILLIERTE BESCHREIBUNG DER ERFINDUNG

Im Folgenden soll die zentrale Idee der erfindungsgemäßen Lösung anhand der in Fig. 1 bis 3 abgebildeten Ausführungsbeispiele näher erläutert werden. Die Bedeutung der mit Bezugszeichen versehenen Symbole in Fig. 1 bis 3 kann der beigefügten Bezugszeichenliste entnommen werden.

10 Im Rahmen eines Ausführungsbeispiels der vorliegenden Erfindung ist ein zur On-Board-Programmierung und In-System-Konfiguration (ISC) eines Flash-Speichers 116 geeignet angepasstes ASIC 108 vorgesehen, das über eine JTAG-Schnittstelle 102 gemäß IEEE-Standard 1149.1 ansteuerbar ist und den Flash-Speicher 116 auch während des Betriebs neu programmiert bzw. konfiguriert. Erfindungsgemäß können dabei das ASIC 108 und der zu programmierende Flash-Speicher 116 im Rahmen des IEEE-Standards 1532 als zusammenhängende Einheit 120 betrachtet werden (siehe Fig. 1 und 2). Demzufolge ist es unerheblich, ob sich der Flash-Speicher 116 innerhalb oder außerhalb des ASICs 108 befindet.

Verleiht man diesem ASIC 108 Eigenschaften gemäß IEEE-Standard 1532, kann der Flash-Speicher 116 auf die gleiche Art wie ein programmierbarer Logikbaustein (PLD) programmiert werden. Dabei werden die Daten des Stromlaufplans bzw. der daraus ableitbaren Netzliste, welche die Konfiguration der Schnittstelle 118 und 119b zwischen dem zu programmierenden Flash-Speicher 116 und dem ASIC 108 festlegen, im BSDL-File abgespeichert. Daraus ergeben sich die folgenden Vorteile:

- Da das BSDL-File jetzt alle Informationen für die Programmierung des Flash-Speichers 116 enthält, muss der Stromlaufplan bzw. die daraus ableitbare Netzliste für die Programmiererstellung nicht mehr herangezogen werden. Somit kann der Programmieralgorithmus wie bei programmierbaren Logikbausteinen (PLDs) automatisch von JTAG-Tools, wie z.B. dem BSCAN-Softwarepaket SYSTEM CASCON™ der Firma GÖPEL electronic GmbH, generiert werden.

10

- Ferner ist eine simultane Programmierung und/oder Konfiguration mehrerer, auf derselben Leiterplatte 100 montierter Halbleiterbausteine 114, 116 möglich, wodurch bereits bei der Produktion der Leiterplatte 100 eine erhebliche Kostenreduzierung erreicht werden kann.

15

- Darüber hinaus ermöglicht die zugrunde liegende Erfindung nach Anschluss eines Configuration-Registers 119b zur Zwischenspeicherung anfallender Adress- und Steuerdaten an die Ausgänge des zur Programmierung des Flash-Speichers 116 vorgesehenen ASICs 108 auch die Programmierung bzw. Konfiguration des Flash-Speichers 116 bei Burstbetrieb.

20

Der IEEE-Standard 1532 lässt mehrere Optionen für die Bereitstellung von Steuer-, Daten- und Adressinformation über die JTAG-Schnittstelle 102 zu. In einem Ausführungsbeispiel der vorliegenden Erfindung wird das nach IEEE-Standard 1532 spezifizierte „Memory Array“ eines PLDs, welches die Konfigurationsdaten enthält, die die programmierten Funktionen des PLDs festlegen, durch eine parallele Schnittstelle 118, 119b (externes Memory Array) zu einem externen Flash-Speicher 116 ersetzt. Hier bieten sich wiederum mehrere Möglichkeiten an:

25

30

- Die parallele Schnittstelle zwischen dem ASIC 108 und dem zu programmierenden Flash-Speicher 116 wird durch eine Serienschaltung mehrerer Speicherzellen 204a, die Teil des BSCAN-Registers 118 sind, gebildet (siehe Fig. 1).

5

- Alternativ kann ein zur In-System-Konfiguration (ISC) des zu programmierenden Speicherbausteins 116 geeignet angepasstes Configuration-Register 119b, wie z.B. eines der gemäß IEEE-Standard 1532 spezifizierten Register ISC_PData bzw. ISC_RData, an die Ausgänge des ASICs 108 als parallele Schnittstelle zwischen dem ASIC 108 und dem zu programmierenden Flash-Speicher 116 angeschlossen werden (siehe Fig. 2). Wie in Fig. 3 dargestellt, sind dann wahlweise drei verschiedene Verbindungen möglich:

15

a) eine Verbindung der Steuer-, Daten- und/oder Adress-
eingänge des zu programmierenden Flash-Speichers 116
mit dem ASIC 108 (Normalbetrieb),

20

b) eine Verbindung der Steuer-, Daten- und/oder Adress-
eingänge des zu programmierenden Flash-Speichers 116
mit einem Signaleingang für BSCAN-Testdaten (SCAN_IN)
zur Funktionsprüfung des Flash-Speichers oder

25

c) eine Verbindung der Steuer-, Daten- und/oder Adress-
eingänge des zu programmierenden Flash-Speichers 116
mit dem Configuration-Register 119b zur Konfiguration
des Flash-Speichers mit Hilfe der im BSDL-File abge-
speicherten Konfigurationsdaten.

30

Dies wird durch Modifikation der BSCAN-Zelle 204 erreicht,
wie im Fig. 3 dargestellt. Erfindungsgemäß verfügt eine sol-
che BSCAN-Zelle 204 über einen Eingangs-Multiplexer 302a zur

Durchschaltung eines von mindestens zwei Eingangssignalen (DATA_IN, SCAN_IN) in Abhängigkeit eines an einem Steuersignaleingang (SEL) anliegenden Steuersignals (SHIFT_DR), wodurch wahlweise eine Verbindung vom ASIC 108 bzw. von einem

5 Signaleingang für BSCAN-Testdaten (SCAN_IN) zu dem Flash-Speicher 116 durchgeschaltet wird. Ein Scan- bzw. Capture-Flipflop 304 dient zur Zwischenspeicherung der von dem ASIC 108 erhaltenen Programmierdaten (DATA_IN) bzw. der BSCAN-Testdaten (SCAN_IN). Zur Auslösung bzw. zur Beendigung einer
10 Schreiboperation wird ein Update-Flipflop 306 verwendet, mit dessen Hilfe einzelne Steuersignal-, Daten- und/oder Adress-eingänge des Flash-Speichers 116 angesteuert werden. Über einen Ausgangs-Multiplexer 302b1 bzw. 302b2 zur Durchschaltung eines von mindestens zwei Eingangssignalen (DATA_IN, SCAN_

15 OUT, Conf data) kann in Abhängigkeit eines an einem Steuer-sig-naleingang (SEL) anliegenden Steuersignals (MODE) wahlwei-se eine Verbindung vom ASIC 108 bzw. von einem Signaleingang für Konfigurationsdaten (Conf data) zu dem Flash-Speicher 116 durchgeschaltet werden.

20

Eine Scan- bzw. Capture-Zelle 204a des Configuration-Registers 119b zur In-System-Konfiguration (ISC) des Flash-Speichers 116 besteht nach einem ersten Ausführungsbeispiel der vorliegenden Erfindung aus einem Scan- bzw. Capture-
25 Flipflop 308 zur Zwischenspeicherung der vom ASIC 108 erhaltenen Konfigurationsdaten (Conf data) und einem Ausgangs-Multiplexer 302c zur Durchschaltung eines von zwei Eingangssignalen (SCAN_OUT, Conf data) in Abhängigkeit eines an einem Steuersignaleingang (SEL) anliegenden Steuersignals (UPD_DR).

30 Somit kann wahlweise eine Verbindung vom ASIC 108, von einem Signaleingang für BSCAN-Testdaten (SCAN_IN) bzw. von einem Signaleingang für Konfigurationsdaten (Conf data) zu dem Flash-Speicher 116 durchgeschaltet werden.

Gemäß einem zweiten Ausführungsbeispiel der vorliegenden Erfindung wird anstelle des Ausgangs-Multiplexers 302c ein Update-Flipflop 306 zur Ansteuerung einzelner Steuersignal-,
5 Daten- und/oder Adresseingänge des Speicherbausteins 116 zwecks Auslösung bzw. Beendigung einer Schreiboperation verwendet.

Der entscheidende Vorteil der vorliegenden Erfindung gegenüber
10 über herkömmlichen On-Board-Programmierverfahren nach dem Stand der Technik besteht darin, dass mit Hilfe des erfindungsgemäßen Verfahrens die Unterschiede zwischen Flash- und
PLD-Programmierung eliminiert werden. Dabei bilden ASIC 108
und Flash-Speicher 116 eine Einheit im Rahmen des IEEE-
15 Standards 1532. Dadurch dass die Stromlaufeigenschaften im BSDL-File gespeichert sind, ergibt sich für den Anwender auch
der Vorteil einer erheblichen Verringerung des Programmieraufwands, da die Verbindungen zwischen dem ASIC 108 und dem
Flash-Speicher 116 nicht mehr aus dem Stromlaufplan bzw. der
20 daraus ableitbaren Netzliste entnommen werden müssen.

Patentansprüche

1. Verfahren zur On-Board-Programmierung und/oder In-System-
5 Konfiguration (ISC) eines Speicherbausteins (116) auf einer
Leiterplatte (100, 200) durch Ansteuerung der Eingänge des
Speicherbausteins (116) mit Hilfe eines auf der Leiterplatte
(100, 200) montierten Halbleiterbausteins (108) über einzelne
Speicherzellen (204) eines Boundary Scan (BSCAN)-Registers
10 (118) zur Aktivierung bzw. Deaktivierung einer Schreibopera-
tion, wobei
- die Architekturbeschreibung des Halbleiterbausteins (108)
und des zu programmierenden Speicherbausteins (116) sowie
das Datenformat der Programmier- (DATA_IN) und Konfigurati-
15 onsdaten (Conf data) in einer Datei bereitgestellt werden,
und
 - die Leiterplatte (100, 200) über eine Schnittstelle (102)
zur Ein- bzw. Ausgabe von Standardbussignalen (TDI, TDO,
TMS, TCK und TRST) sowie zur Eingabe der Ansteuersignale
20 des Halbleiterbausteins (106) ansteuerbar ist,
dadurch gekennzeichnet, dass
- weiterhin die Daten der Netzliste bzw. des Stromlaufplans,
welche(r) die Konfiguration der Schnittstelle (102) zwischen
dem zu programmierenden Speicherbaustein (116) und dem Halb-
25 leiterbaustein (108) festlegt, in weiteren Dateien bereitge-
stellt werden.
2. Verfahren nach Anspruch 1,
dadurch gekennzeichnet, dass
- 30 der Halbleiterbaustein (108) und der zu programmierende Spei-
cherbaustein (116) als zusammenhängende Einheit (120) ange-
steuert werden.

3.. Verfahren nach einem der Ansprüche 1 oder 2,
dadurch gekennzeichnet, dass

der Programmieralgorithmus zur Programmierung des Speicher-
bausteins (116) automatisch durch Zugriff auf die weitere Da-
5 tei erzeugt wird.

4. Verfahren nach einem der Ansprüche 1 bis 3,
gekennzeichnet durch

10 die simultane Programmierung mehrerer Speicherbausteine (114,
116) auf derselben Leiterplatte (100) über die Schnittstelle
(102):

5. Verfahren nach einem der Ansprüche 1 bis 4,
gekennzeichnet durch

15 die Implementierung eines Datenregisters (119b) zur Zwischen-
speicherung der bei Burstbetrieb anfallenden Adress- und
Steuerdaten, angeschlossen an die Ausgänge des für die Pro-
grammierung des Speicherbausteins (116) vorgesehenen anwen-
dungsspezifischen integrierten Halbleiterbausteins (108).

20

6. Parallele Schnittstelle zur On-Board-Programmierung und/
oder In-System-Konfiguration (ISC) eines Speicherbausteins
(116) auf einer Leiterplatte (100, 200) durch Ansteuerung
einzelner Eingänge des Speicherbausteins (116) mit Hilfe ei-
25 nes auf der Leiterplatte (100, 200) montierten anwendungsspe-
zifischen integrierten Halbleiterbausteins (108) über Spei-
cherzellen (204) eines Boundary Scan (BSCAN)-Registers (118)
zur Aktivierung bzw. Deaktivierung einer Schreiboperation,
dadurch gekennzeichnet, dass

30 die parallele Schnittstelle durch eine Serienschaltung mehrerer Speicherzellen (204a), die Teil des BSCAN-Registers (118) sind, gebildet ist.

7. Parallele Schnittstelle zur On-Board-Programmierung und/oder In-System-Konfiguration (ISC) eines Speicherbausteins (116) auf einer Leiterplatte (100, 200) durch Ansteuerung einzelner Eingänge des Speicherbausteins (116) mit Hilfe eines auf der Leiterplatte montierten anwendungsspezifischen integrierten Halbleiterbausteins (108) über Speicherzellen (204) eines Boundary Scan (BSCAN)-Registers (118) zur Aktivierung bzw. Deaktivierung einer Schreiboperation, dadurch gekennzeichnet, dass
- 10 die parallele Schnittstelle durch Serienschaltung mehrerer Speicherzellen (204a) eines zur In-System-Konfiguration (ISC) des zu programmierenden Speicherbausteins (116) verwendeten Datenregisters (119b) gebildet ist.
- 15 8. Speicherzelle eines Boundary Scan (BSCAN)-Registers (118) zur On-Board-Programmierung eines Speicherbausteins (116) auf einer Leiterplatte (100, 200) durch Ansteuerung einzelner Eingänge des Speicherbausteins (116) mit Hilfe eines auf der Leiterplatte (100, 200) montierten anwendungsspezifischen integrierten Halbleiterbausteins (108) zur Aktivierung bzw. Deaktivierung einer Schreiboperation, gekennzeichnet durch
- 20 - einen Eingangs-Multiplexer (302a) zur Durchschaltung eines von mindestens zwei Eingangssignalen (DATA_IN, SCAN_IN) in Abhängigkeit eines an einem Steuersignaleingang (SEL) anliegenden Steuersignals (SHIFT_DR), wodurch wahlweise eine Verbindung vom Halbleiterbaustein (108) bzw. von einem Signaleingang für BSCAN-Testdaten (SCAN_IN) zu dem Speicherbaustein (116) durchgeschaltet wird,
- 25 - ein Scan- bzw. Capture-Flipflop (304) zur Zwischenspeicherung der vom Halbleiterbaustein (108) erhaltenen Programmierdaten (DATA_IN) bzw. der BSCAN-Testdaten (SCAN_IN),
- 30

- ein Update-Flipflop (306) zur Ansteuerung einzelner Steuerungssignal-, Daten- und/ oder Adresseingänge des Speicherbausteins (116) zur Auslösung bzw. Beendigung einer Schreiboperation und
 - 5 - einen Ausgangs-Multiplexer (302b1, 302b2) zur Durchschaltung eines von mindestens zwei Eingangssignalen (DATA_IN, SCAN_OUT, Conf data) in Abhängigkeit eines an einem Steuerungssignaleingang (SEL) anliegenden Steuersignals (MODE), wodurch wahlweise eine Verbindung vom Halbleiterbaustein
 - 10 (108) bzw. von einem Signaleingang für Konfigurationsdaten (Conf data) zu dem Speicherbaustein (116) durchgeschaltet wird.
9. Speicherzelle eines Datenregisters (119b) zur In-System-
- 15 Konfiguration (ISC) eines Speicherbausteins (116) auf einer Leiterplatte (100, 200) durch Ansteuerung einzelner Eingänge des Speicherbausteins (116) mit Hilfe eines auf der Leiterplatte (100, 200) montierten anwendungsspezifischen integrierten Halbleiterbausteins (108) zur Aktivierung bzw. Deaktivierung einer Schreiboperation,
- 20 gekennzeichnet durch
- ein Scan- bzw. Capture-Flipflop (308) zur Zwischenspeicherung der vom Halbleiterbaustein (108) erhaltenen Konfigurationsdaten (Conf data) und
- 25 - einen Ausgangs-Multiplexer (302c) zur Durchschaltung eines von zwei Eingangssignalen (SCAN_OUT, Conf data) in Abhängigkeit eines an einem Steuerungssignaleingang (SEL) anliegenden Steuersignals (UPD_DR), wodurch wahlweise eine Verbindung vom Halbleiterbaustein (108), von einem Signaleingang
- 30 für BSCAN-Testdaten (SCAN_IN) bzw. von einem Signaleingang für Konfigurationsdaten (Conf data) zu dem Speicherbaustein (116) durchgeschaltet wird.

10. Speicherzelle nach Anspruch 9,
dadurch gekennzeichnet, dass
anstelle des Ausgangs-Multiplexers (302c) ein Update-Flipflop
(306) zur Ansteuerung einzelner Steuersignal-, Daten- und/
5 oder Adresseingänge des Speicherbausteins (116) zwecks Auslö-
sung bzw. Beendigung einer Schreiboperation verwendet wird.

Zusammenfassung

Programmierung eines Speicherbausteins über ein Boundary Scan-Register

5

Die Erfindung basiert auf einem Verfahren sowie einer parallelen Schnittstelle (118, 119b) zur On-Board-Programmierung und/oder In-System-Konfiguration (ISC) eines auf einer Leiterplatte (100, 200) montierten Speicherbausteins (116) durch
10 Ansteuerung seiner Eingänge mit Hilfe eines auf derselben Leiterplatte (100, 200) montierten anwendungsspezifischen integrierten Halbleiterbausteins (108) über ein Boundary Scan (BSCAN)-Register (118), dessen Ausgangssignale zur Aktivierung bzw. Deaktivierung einer Schreiboperation vorgesehen
15 sind. Dabei sind die Architekturbeschreibung des Halbleiterbausteins (108) und des zu programmierenden Speicherbausteins (116) sowie das Datenformat der Programmier- (DATA_IN) und Konfigurationsdaten (Conf data) in einem Boundary-Scan Description Language (BSDL)-File abgespeichert. Die Leiterplatte
20 (100, 200) ist über eine zur Durchführung einer Funktionsprüfung des Speicherbausteins (116) geeignete JTAG-Schnittstelle (102) zur Ein- bzw. Ausgabe von Standardbussignalen (TDI, TDO, TMS, TCK und TRST) sowie zur Eingabe der Ansteuersignale des Halbleiterbausteins (106) ansteuerbar.

25

Zur Verringerung des Programmieraufwands werden erfindungsgemäß die Daten des Stromlaufplans bzw. der daraus ableitbaren Netzliste, welche die Konfiguration der Schnittstelle (118, 119b) zwischen dem zu programmierenden Speicherbaustein (116)
30 und dem Halbleiterbaustein (108) festlegen, in dem BSDL-File abgespeichert, so dass der zu programmierende Speicherbaustein (116) nach Anpassung der Eigenschaften des Halbleiterbausteins (108) an den IEEE-Standard 1532 über eine JTAG-

Schnittstelle (102) gemäß IEEE-Standard 1149.1 in gleicher Weise wie ein programmierbarer Logikbaustein (PLD) programmiert werden kann.

5 Figuren 1 und 2

100

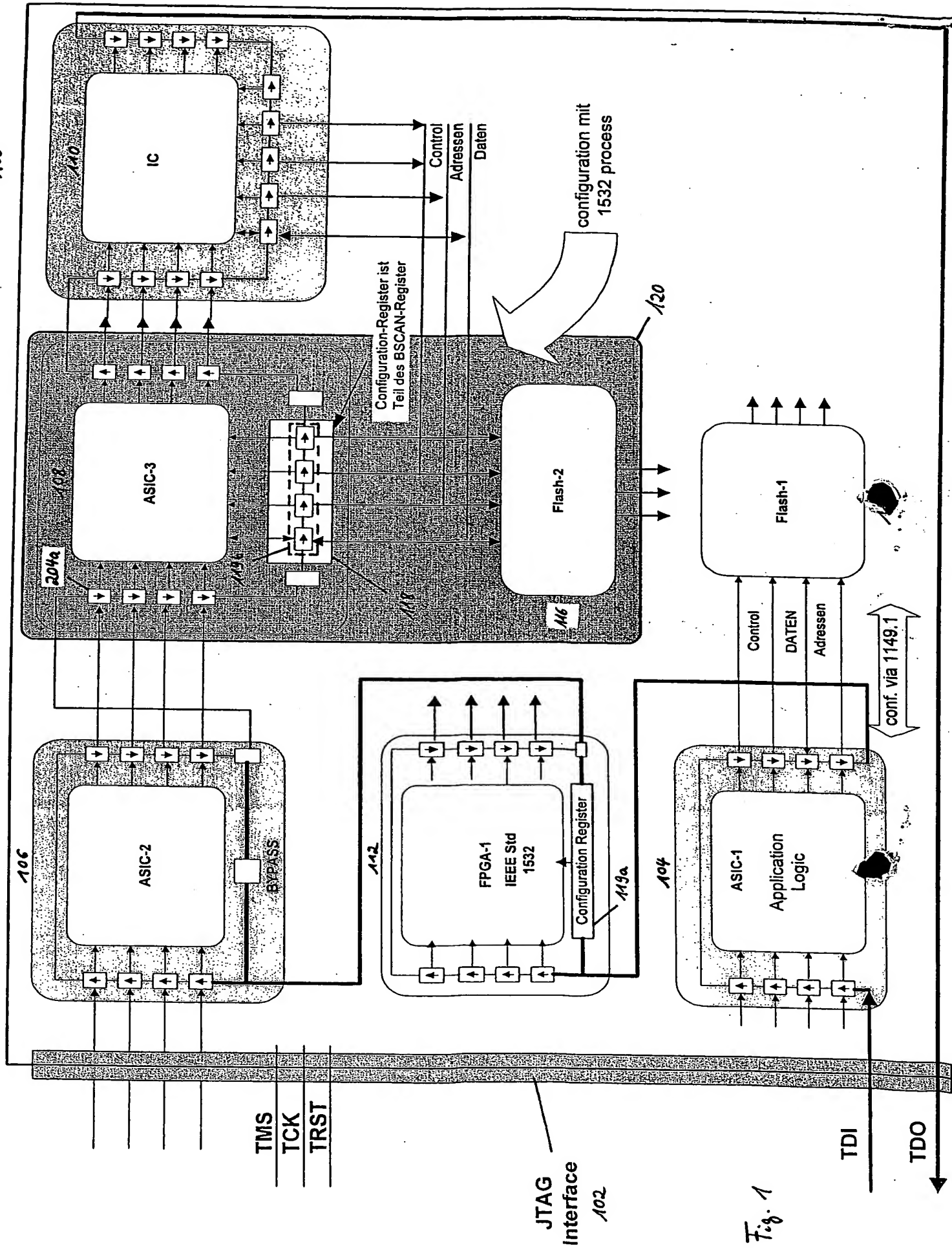


Fig. 1

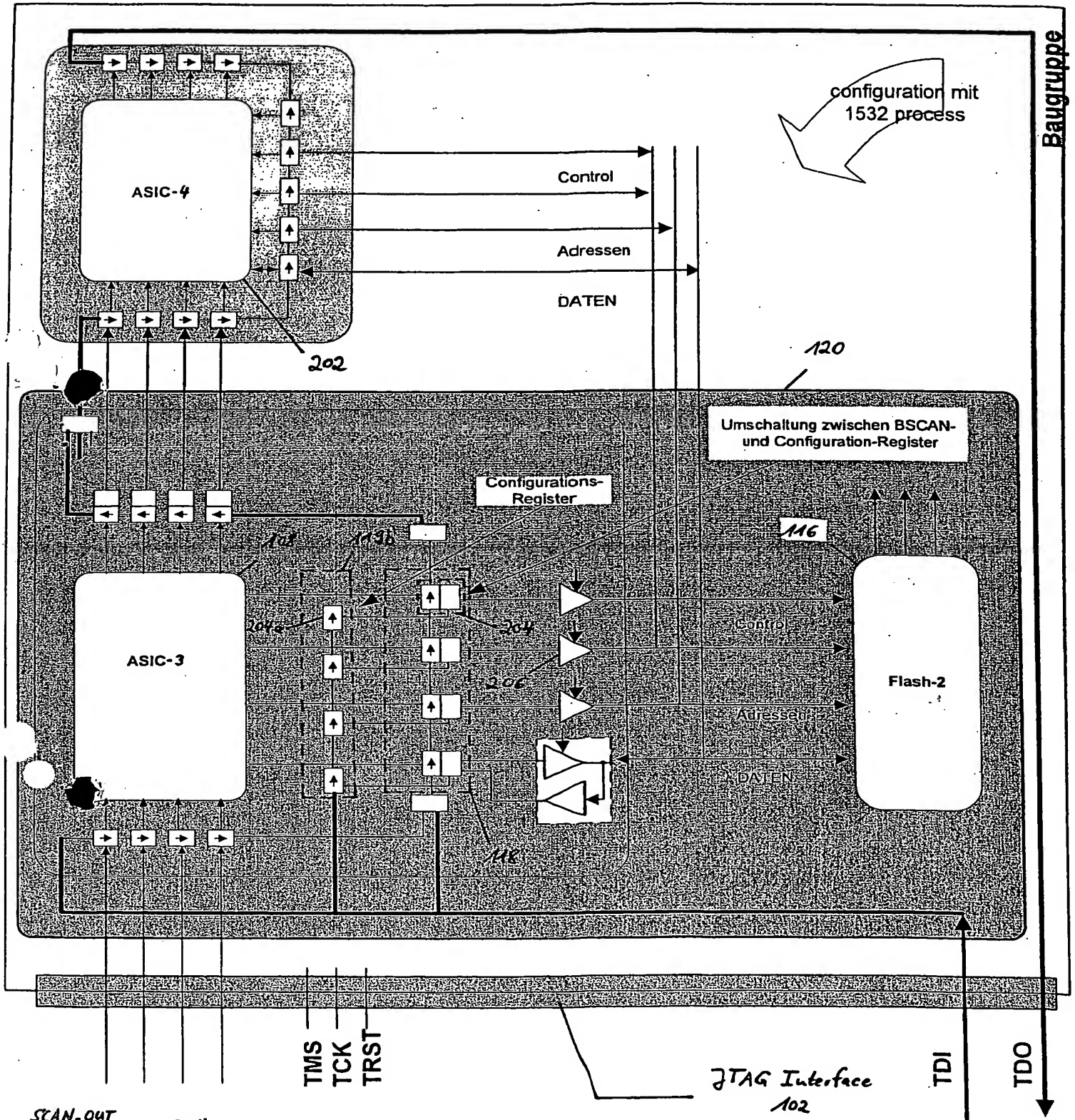
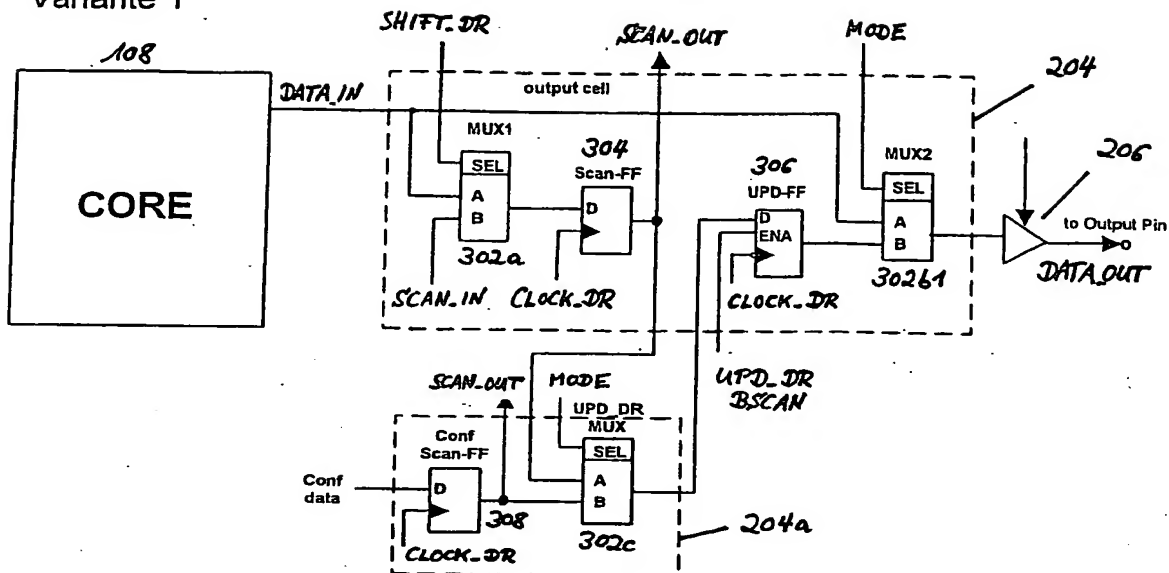
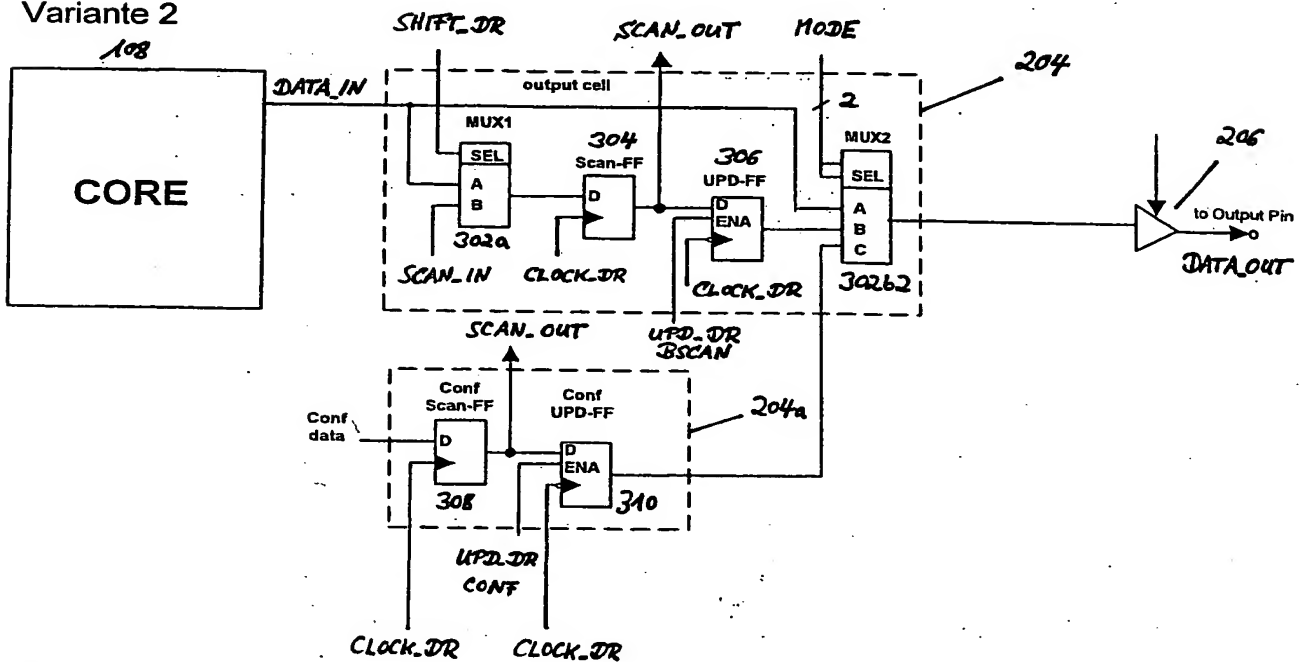


Fig. 2

Variante 1



Variante 2



Variante 1: Multiplexer vor UPD-FF

Fig. 3

Variante 2:

2. UPD-FF fuer Configuration-Data und Erweiterung des Output-Multiplexers